Statistical Methods in AI (CSE/ECE 471)
Spring-2020
Assignment-2 (550 points)
Posted on: 16/02/20
Due on: 11:55 P.M. 04/03/20

Instructions

1. The assignment contains six questions. All the questions are compulsory

2. Assignment must be implemented in Python 3 only.

3. You are allowed to use libraries for data preprocessing (numpy, pandas etc.) and for evaluation metrics, data visualization (matplotlib etc.).

4. You will be evaluated not just on the overall performance of the model on the test set but also on the experimentation with hyper parameters, data prepossessing techniques etc.

5. Datasets for all the questions are provided at http://bit.ly/smai_a2_data

6. For each question, you are required to make two files - one for the code and the second for the report.

7. The report file must be a well documented jupyter notebook, explaining the experiments you have performed, evaluation metrics and corresponding code. The code must run and be able to reproduce the accuracies, figures/graphs etc.

8. For all the questions except clustering , you must create a train-validation data split and test the hyperparameter tuning on the validation set . Your jupyter notebook must reflect the same.

9. Your assignment will be evaluated with undisclosed test files.

10. Your final submission folder should be named ”RollNo.zip”. This should contain a single folder ”RollNo” that has 13 files - q1.py-q6.py and q1.ipynb - q6.ipynb,test.py. Do not include any other files. Strictly adhere to the naming convention.

11. Any attempts at plagiarism will be penalized heavily.

Questions

1. (100 points) **Image Classification**

   1. Given CiFAR-10 dataset, implement a linear SVM classifier to predict the classes of the test images.
   2. Featurize the images as vectors that can be used for classification.
   3. Report your observations for different values of C. Explain the significance of C.
   4. Compare and contrast the classifier with the KNN classifier built in the previous assignment.
   5. Report accuracy score, F1-score, Confusion matrix and any other metrics you feel useful.
   6. Report the support vector images in each case.
   7. (**Bonus-20 points**) You may do some processing on the train set to improve your scores on linear SVM. Report your changes clearly.
   8. You can use **inbuilt** functions for SVM.

2. (100 points) **Gaussian Mixture Models Clustering**

   1. You are given 3 data files(dataset1.pkl,dataset2.pkl,dataset3.pkl) and 1 code file gmm.py. The code consists of -
      (a) Function to load dataset.
      (b) Function to save dataset.
      (c) Class GMM1D which consists multiple functions.
   2. Load dataset .
   3. Use inbuilt sklearn functions to cluster(GMM clustering) the points and plot them. Also report no of iterations taken to converge.
   4. In GMM1D, fill in the blanks with code and cluster the points. Plot for **each** iteration.
   5. (**Bonus-20 points**) Plot the log likelihood graph to show the behaviour.

3. (75 points) **Linear Regression**

   1. Given a NASA data set, obtained from a series of aerodynamic and acoustic tests of two and three-dimensional airfoil blade sections. Implement a linear regression model from scratch using gradient descent to predict scaled sound pressure level. The various attributes of the data are explained in the file `description.txt`.
   2. Using appropriate plot show how number of iterations is affecting the mean squared error for above model under below given conditions:
      (a) Using 3 different initial regression coefficients (weights) for fixed value of learning parameter (All 3 in single plot).

(b) Using 3 different learning parameters for some fixed initial regression coefficients. (All 3 in single plot)

3. If you want to apply regression on some dataset but one of it's features has missing values under below given conditions, how will you approach the problem. (No need of Code Experimentation)

   (a) When 0-0.5% of values are missing of that feature

   (b) When 8-10% of values are missing of that feature

   (c) When 60-70% of values are missing of that feature

4. (75 points) **Linear Regression**

   1. Given a dataset containing historical weather information of certain area, implement a linear regression model from scratch using gradient descent to predict the apparent temperature. The various attributes of the data are explained in the file `description.txt`. Note that attributes are text, categorical as well as continuous. **Note:** Test data will have 10 columns. Apparent temperature column will be missing from in between.
   2. Compare the performance of different error functions (Mean square error, Mean Absolute error, Mean absolute percentage error) and explain the reasons for the observed behaviour.
   3. Analyse and report the behaviour of the regression coefficients(for example: sign of coefficients, value of coefficients etc.) and support it with appropriate plots as necessary.

5. (100 points) **Support Vector Machine**

   1. Given a dataset which contains a excerpts of text written by some author and the corresponding author tag, implement an SVM classifier to predict the author tag of the test text excerpts.
   2. For the feature extraction of the text segments, either use Vectorizers provided in sklearn or use pre-trained word embedding models. (Code snippet for usage of word embedding models is given here).
   3. Visualize the feature vectors and see if you could find some pattern.
   4. Tweak different parameters of the Linear SVM and report the results.
   5. Experiment different kernels for classification and report the results.
   6. Report accuracy score, F1-score, Confusion matrix and any other metrics you feel useful.
   7. (**Bonus-20 points**) You may do some pre-processing on textual data to improve your classifier. Explain why score has improved if it did.
   8. Link to the dataset has been provided in the common link.
   9. You can use **inbuilt** functions for SVM.

10. The code file must be a python(.py) file. You are expected to define a class for each question which is compatible with the test.py file provided here. Make sure your code can be run by ”python test.py”. Double check this.

6. (100 points) **Clustering**

   1. Given a dataset of documents with content from 5 different fields (namely business, entertainment, politics, sport, and tech), cluster them using any clustering algorithm of your choice.
   2. Do not use any libraries for this part. You are expected to code your clustering algorithm from scratch.
   3. For feature extraction you can use the vectorizers provided by sklearn or by using the pre trained embeddings. (Code snippet for the usage of these embeddings has been provided in the previous question).
   4. You might have to perform some pre-processing on the raw documents before you apply your algorithm.
   5. We have provided ground truth document tags for the documents. Report accuracy score on these documents.
   6. We will test your score on the documents for which the tags have not been provided.
   7. In the dataset, the number after the '_' symbol in the file name denotes the cluster label.
   8. The code file must be a python(.py) file. You are expected to define a class for each question which is compatible with the test.py file provided here. Make sure your code can be run by ”python test.py”. Double check this.

All the best!!!!